An Adaptive Raptor Enabled Data Carousel for File Delivery in 802.11 Multicast Networks

Berna Bulut, Angela Doufexi and Andrew Nix

Communication Systems & Networks, Department of Electrical and Electronic Engineering

University of Bristol, Bristol, United Kingdom

Email: {berna.bulut; a.doufexi; andy.nix}@bristol.ac.uk

***Abstract*—WLANs currently do not provide a robust solution for reliable data multicast. Multicast packets are delivered to multiple users as a simple broadcast service without support for Automatic Repeat Request. Hence, a fixed low speed transmission mode is generally used to improve the reliability of multicast files. However, this results in the inefficient use of bandwidth. This paper proposes a reliable and efficient Wi-Fi multicast delivery solution for use in challenging outdoor environments. We proposed an adaptive Application Layer Forward Error Correction (AL-FEC) enabled data carousel for reliable multicast transmission over standard 802.11 WLANs. To quantify the benefits of the proposed system, results are presented from a cross-layer simulator combining novel outdoor ray-tracing, a physical layer abstraction simulator and a RaptorQ enabled multicast data carousel simulator. The simulation results demonstrate that adaptive FEC carousels significantly reduce the average download time, increase the percentage of satisfied users and bandwidth efficiency in a multicast network.**

Keywords— carousels; RaptorQ; reliable multicast; IEEE 802.11 WLAN

I. INTRODUCTION

The wide availability of cell phones and tablet computers has led to an increase in the demand for mobile multimedia applications. Unicast protocols struggle to meet these demands since scarce radio and network resources are shared between users. One solution is to efficiently disseminate high bandwidth media-rich content over error-prone wireless channels to many users in the form of multicast transmissions. However, robust and reliable data transmission is incompatible with multicast 802.11 transmissions. For unicast transmissions each user is sent a unique copy of the media. As a consequence, for high user densities the network rapidly runs out of bandwidth. The problem is made even worse since each unicast user must also request the retransmission of any lost or corrupted data packets via the return channel. While reliability is achieved, it prevents the dissemination of media rich content to large user groups.

At present 802.11 offers no standardized or certified extension for reliable multicast delivery. Multicast packets are transmitted as a simple broadcast service without support for Automatic Repeat Request. Another issue with multicast transmission over 802.11 networks is the lack of link adaptation. Adaptive modulation and coding is not supported for multicast groups. In practice, to improve reliability, multicast transmission often occurs using the *lowest* 802.11 link speed regardless of channel conditions. This approach is *very* wasteful of valuable radio spectrum.

In scenarios where a return channel is unavailable, it is well known that a data carousel or broadcast disk [1] approach can be used to provide reliable multicast. With a data carousel the transmitter continually transmits all data packets in a cyclic fashion. Receivers may join the carousel at any time and normally leave only when they have received all the packets that belong to the desired file(s). However, wireless communication channels are prone to errors (which result in lost packets) and as a consequence users may not obtain all elements of the required file(s) in a single transmission cycle. In such cases the users must wait for the next carousel cycle for the chance to successfully retrieve the file. This approach may result in numerous duplicate packets at the receiver and a significant increase in the total time required to acquire the desired media. Application Layer Forward Error Correction (AL-FEC) based on traditional block codes can be used in conjunction with data carousels (called FEC carousel) to reduce download delay, as reported in [2], [3], [4]. However, traditional codes suffer from constraints such as a fixed code rate that must be defined beforehand. Furthermore, prior knowledge of the channel conditions is required. This approach may still result in the observation of duplicate packets at the receiver, if the code rate is underestimated, i.e. limited redundant packets.

Raptor codes [5] are a form of fountain code that can generate on-the-fly an unlimited number of encoded symbols from a fixed source block. A Raptor code rate can be adjusted dynamically according to the channel conditions. Due to this property, these codes are characterized as rateless codes. In this case all received symbols at the receiver are different and hence useful in the decoding process. In a fountain code, it does not matter which particular symbols are received, so long as a *sufficient* number of symbols arrive. In this case the file can be decoded at the receiver. This property makes Raptor codes desirable for carousel-based services since the probability of receiving duplicate symbols can be reduced significantly. Our previous works in [6], [7] proposed an AL-FEC (based on RaptorQ (RQ) codes [8]) enhanced data carousel for use in outdoor 802.11 multicast wireless LANs.

Field trials were also conducted to demonstrate and quantify the system performance [9]. Results showed that the proposed system enhances users quality of experience (QoE) while efficiently utilise the available bandwidth since higher modulation and coding schemes (MCS) become viable by exploiting the rateless properties of RaptorQ codes. In previous work, fixed MCS modes were used during the multicast transmission. However, the network utilisation and users QoE can be further enhanced, if the multicast MCS mode is selected according to the channel condition of the users within the coverage area. Therefore, in this paper, an adaptive FEC carousel system based on Raptor codes is proposed. First, we formulated the design problem and then we have developed a cross-layer simulator in order to evaluate the performance of this system in a challenging and realistic outdoor environment. The percentage of satisfied users and the average download time are used as key performance metrics.

The remainder of this paper is organized as follows. Section II explains the details of cross-layer simulation. The proposed system is explained in Section III. Simulation results and analysis are provided in Section IV with conclusions presented in Section V.

II. Cross-Layer Simulation

A cross-layer simulator has been developed by the authors to evaluate the end-to-end system performance. In order to reduce the computational complexity, the overall system is divided into modular subsystems (FEC carousel, Raptor, Wi-Fi MAC-PHY layer and channel simulator), each of which is modelled independently.

A. FEC carousel model and Raptor codes

The model can be summarised as follows. A file is divided into source blocks, with these blocks further divided into k source symbols with T bytes. A systematic RQ encoder is then applied to each individual source block of the file to generate the encoded data. The partitioning process and transmission schedule used in our RQ software is shown in Fig. 1. Every time the RQ encoder generates one encoded symbol from each source block of every file. As shown in Fig. 1, the first encoded symbol from each of the source blocks is transmitted, followed by the second and so on. In this case, symbols of each block and file are interleaved over time. As the code is systematic the first k encoded symbols are the source (original) symbols. RQ can be used to generate 2^{24} repair symbols from a source block therefore the maximum number of repair symbols in the software was set to2^{24}. Hence, each time a new encoded symbol is transmitted for each source block in order to avoid duplicate packets at the receiver. It is assumed that one encoded symbol is placed into one UDP/IP packet and also that 802.11 multicast/broadcast packets cannot be fragmented at the MAC layer. Hence, there is a 1:1 mapping between encoded symbol and PHY layer Protocol Data Units (PPDU). Therefore, packet error rate (PER) at MAC layer equals to the PER at the application layer.

At the receiver the Raptor decoder waits to collect all the UDP packets belong to a given source block. If the total number of received symbols k' (source and repair) for a block is $k' \geq k + \alpha$,where α is the number of extra symbols (overhead) required for successful decoding, the Raptor decoder is able to decode and all source packets are recovered and delivered to application layer. However, if the decoder fails, the receiver waits for more packets until successful decoding.

For the FEC carousels the ratio between the numbers of encoded symbols sent in each cycle N_{enc} (i.e. in the first cycle, N_{enc} equals to source plus repair after that only the repair symbols) and the number of source symbols k is called the stretch factor $SF = \frac{N_{enc}}{k}$.

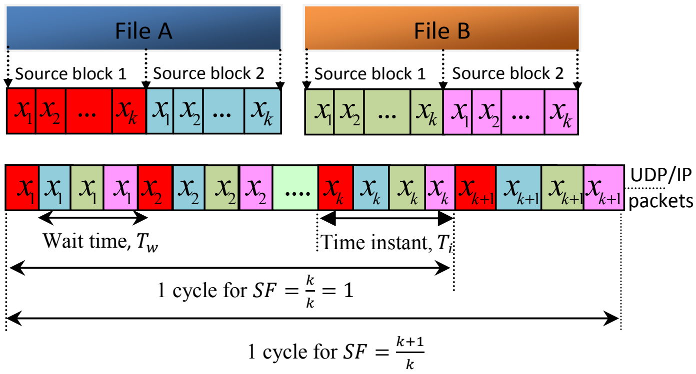

Fig. 1. Carousel with Raptor code (FEC carousel).

A. System and Channel model

As part of the AIYP (Arkive In Your Pocket) project, this paper contributes to the development of a next generation multimedia broadcast system that will radically enhance the experience of a trip to the Zoo by offering a location dependent Wi-Fi application to hundreds of visitors.

The Wi-Fi performance was evaluated in our trial location using a geographic model of Bristol Zoo. Furthermore, we modelled 100 users walking along the routes shown in Fig. 2. The users were served by three access points (APs), which operate in the 2.4GHz band and use a transmit power to the antenna port of 20dBm. The AP locations were optimised to provide higher coverage while using minimum number of APs. The detailed information can be found in our previous work [7].

In this study the user terminal and AP are assumed to use single antenna. A state-of-the-art outdoor 3D ray-tracer [10] was used to model the time varying channel matrix **H** between the AP and each user equipment (UE) in order to replicate the time correlated nature of the received signal as the users move along the site. The ray-tracer makes use of the physical laws of radiowave propagation, such as reflection, diffraction and scattering and identifies all significant ray paths between the AP and the UE in 3D space. The ray-tracing database had a resolution of 2 m and included buildings, foliage and terrain data in the Bristol Zoo area. Point-source ray-tracing is conducted from the user to the AP provides information on the amplitude, phase, time delay, Angle of Departure (AoD) and Angle of Arrival (AoA) of each multipath component (MPC).

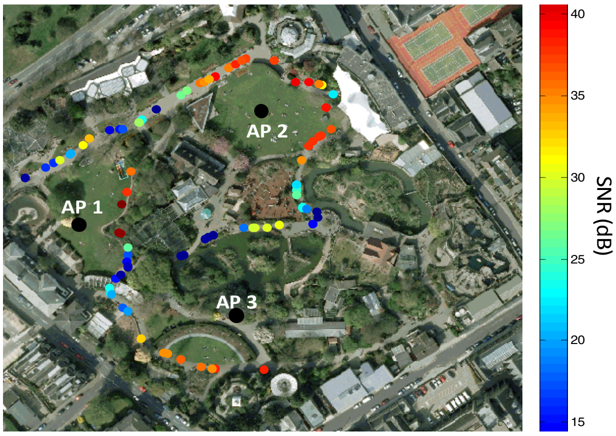

Fig. 2. Locations of APs and possible 100 users with colour-coded received SNRs in the Bristol Zoo.

B. Link-level abstraction(MAC-PHY model)

An Effective SNR Mapping (ESM) PHY abstraction model, known as the Received Bit Mutual Information (RBIR) [11] technique, was used to calculate the PER over time. In the ESM method, a block of OFDM sub-carrier SNRs, which vary severely due to the frequency selective fading, is transformed into a single Effective SNR (ESNR) value. This ESNR value is then used to define the instantaneous PER for any MCS mode using a non-faded PER versus SNR look up table. This table was generated via bit accurate Wi-Fi simulation for an AWGN channel. The transmission modes for an 802.11n 20 MHz channel bandwidth with an 800 ns guard interval (GI) [12] were used in the RBIR simulator can be seen in Table I.

TABLE I
802.11N TRANSMISSION MODES

MCS index	Modulation	Coding rate	N_{DBPS}	Data rate (Mbps)
0	BPSK	1/2	3.25	6.5
1	QPSK	1/2	6.5	13
2	QPSK	3/4	9.75	19.5
3	16-QAM	1/2	13	26
4	16-QAM	3/4	19.5	39
5	64-QAM	2/3	26	52
6	64-QAM	3/4	29.25	58.5
7	64-QAM	5/6	32.5	65

The 802.11n MAC-PHY layer simulator, which is implemented according to standard, models the packet loss pattern for a sequence of packets based on the time varying channel $\mathbf{H}$ created by the ray-tracing model.

C. Simulation parameters

The system parameters are summarized as follows: k=200, T=1400 bytes (symbol=packet size), the total number of files in the carousel is 3 and each file is 5.6 MB in size. We assume that users want to download all the files therefore each file has the same priority. The upper layer headers sizes are 8 bytes for UDP, 20 bytes for IP and 36 bytes for MAC layer. The total upper layer overhead is $L_{hdr} = 8 + 20 + 36 = 64$ bytes in size.

III. SYSTEM DESIGN AND EVALUATION

This section presents an adaptive FEC carousel system where the server (transmitter) transmits the files at an optimum MCS mode and defines the optimum number of encoded symbols sent in each carousel cycle. First, the proposed adaptive system is formulated and then the performance evaluation methodology for download delivery in multicast networks is explained.

A. Adaptive multicast FEC carousel design

In multicast transmission, each user experiences a very different channel condition therefore it is not possible to find an MCS mode and *SF* value that are optimal for all users. In this paper, our algorithm aims to find an MCS mode and *SF* values that provide the best transmission (lower average download time) over all users in the coverage area. We assume that the server collects the received SNR from all users in the coverage area and then based on this information, the server defines the best MCS mode m_j and the SF_{opt} for the next carousel cycle.

For carousel based file delivery, the aim is to reduce the average download time over all users. The download time for a file, when Raptor codes are implemented, depends on the packet error rate (PER) in the source blocks, i.e. the source block with the highest PER since the receiver waits a longer time to collect enough number of packets for that block to successfully decode the whole file. Therefore, we used the average number of received packets per second N to source blocks as a performance evaluation metric to be maximised over all users.

Using the interleaved carousel model, we randomise the PER (prevent burst of errors) amongst the source blocks and therefore the mean and variance of the PER over all source blocks is very close to that seen in Fig. 3.

The estimated average number of received packets per second to source blocks for user i is a function of the channel SNR γ_i, the MCS mode m_j, $m_j \in M$, the source block length k and the symbol size T is defined as

$$N_i^{m_j} = \frac{1}{N_{SB}} \sum_{l=1}^{N_{SB}} \left\lfloor \left(\frac{1-PER_{i,l}}{Tx^{m_j}}\right) \right\rfloor, \quad \forall\, m_j \in M \tag{1}$$

$$Tx^{m_j} = DIFS + T_{BO} + T_{PREAMBLE} + T_{SYM}.N_{SYM} + SIFS \tag{2}$$

$$N_{SYM} = \left\lceil \frac{L_{PSDU}+2.75}{N_{DBPS}^{m_j}} \right\rceil \tag{3}$$

where N_{SB} is the total number of source blocks in a file, Tx^{m_j} is the time required to transmit a PHY layer Protocol Data

Units (PPDU) which is the sum of time required to transmit the preamble, the Protocol Service Data Unit (PSDU) L_{PSDU} and DIFS (distributed inter-frame spacing), SIFS (short frame spacing) and back-off time T_{BO}. All parameters follow the 802.11n standard in [12]. N_{SYM} is the number of OFDM symbols required for transmission of an L_{PSDU}, the number 2.75 comes from the overhead of service and tail bits, and $N_{DBPS}^{m_j}$ is the number of data bytes per OFDM symbol for a given MCS mode m_j (see Table I). L_{PSDU} is the sum of the Raptor symbol size T and the total upper layer headers (UDP/IP/MAC) L_{hdr}, $L_{PSDU} = T + L_{hdr}$.

Fig. 4 shows the average number of received packets per second as a function of MCS mode and SNR.

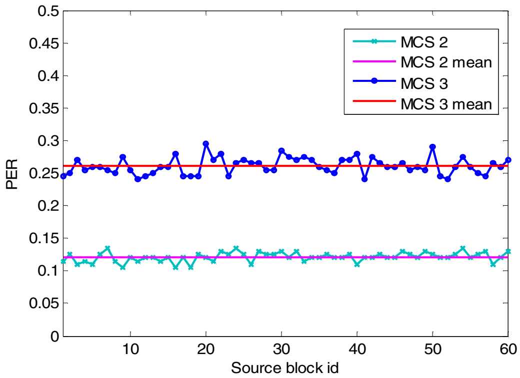

Fig. 3. PER in each source block for MCS mode 2, 3 and SNR=12 dB.

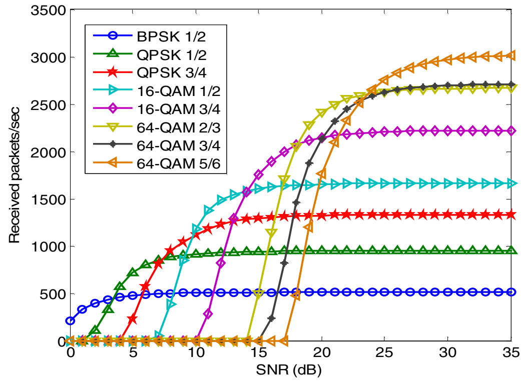

Fig. 4. Average number of received packets per second with respect to SNR.

After calculating, the average number of received packets per second for all K users, we calculate the mean over all these users for each MCS mode m_j,

$$N_{mean}^{m_j} = \frac{1}{K}\sum_{i=1}^{K} N_i^{m_j}, \quad \forall\, m_j \in M,\ \ i = \{1,..,K\} \tag{4}$$

Then, we select the MCS mode m_j that provides the highest average number of received packets per second, such as

$$m_j = \arg\max_{m_j \in M} N_{mean}^{m_j} \tag{5}$$

After choosing the best MCS mode, we calculate the optimum SF value for the selected MCS mode for the next transmission of data. First, for a given SNR γ the number of required packets N_{req} in order to successfully decode the source blocks of a file for each user is calculated as

$$N_{req,i} = \max_{l}\left[\left(\frac{k+\alpha}{1-PER_{i,l}}\right)\right], \quad l = 1, \ldots, N_{SB} \tag{6}$$

The required SF is calculated for each user i such as

$$SF_i = \frac{N_{req,i}}{k}, \quad i = \{1,..,K\} \tag{7}$$

Finally, the optimum SF value SF_{opt}, is chosen such that it allows majority of users to successfully download the file(s) in the first cycle in order to avoid the feedback implosion problem since users that cannot download the file will send a feedback message to the server. The SF_{opt} is given by

$$SF_{opt} = \max_i SF_i\,, \quad i = \{1,..,K\} \tag{8}$$

$$\text{Subject to} \quad SF_i \le SF_{thr}^{m_j}$$

If there are no users with $SF_i \le SF_{thr}$ then SF_{opt} is defined as the maximum SF value over all users $SF_{opt} = \max_i SF_i$ without any SF threshold $SF_{thr}^{m_j}$, constraint.

The $SF_{thr}^{m_j}$ is determined for each MCS mode m_j by calculating the download time versus SNR for a user as seen in Fig. 5. The corresponding required SF values (the maximum value amongst all source blocks) that allow a user to download the files within the first cycle can be seen in Fig. 6. The download time for one user when a fixed MCS mode is used is given by

$$T_D^{m_j} = (N_{SB}.N_F.(SF^{m_j}.k - 1){+}1)\,.Tx^{m_j} \tag{9}$$

where N_F is the number of files in the carousel. Table II shows the $SF_{thr}^{m_j}$ threshold for each MCS mode m_j. For a selected MCS mode m_j, increasing SF, increases the download times for the users that cannot make use of that MCS mode in the current cycle. Therefore, the SF value must be restricted in order not to affect users with bad channel conditions.

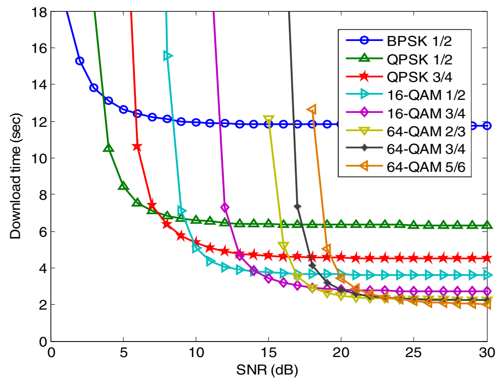

Fig. 5. Download time vs. SNR for a user.

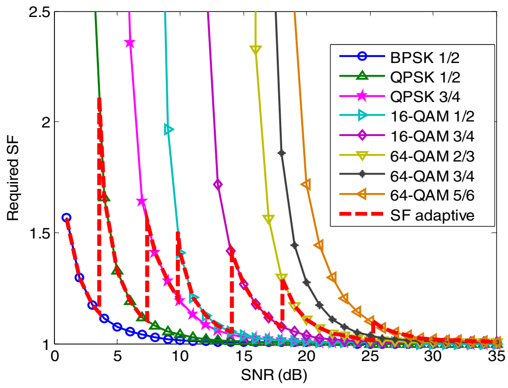

Fig. 6. *SF* values with respect to SNR for a user.

TABLE II
SF THRESHOLD FOR EACH MCS MODE

MCS index	0	1	2	3	4	5	6	7
SF_{thr}	1.56	2.1	1.55	1.5	1.4	1.28	-	1.1

B. Evaluation methodology

The average download time, which is defined as the time elapsed since the user joins the carousel until he has received enough packets to reconstruct the file, and the percentage of satisfied users, which is defined as the number of users that successfully downloaded the files are the key metrics for the overall system performance (service quality) for download delivery in multicast networks in this paper. The objective is to reduce the average download time and increase the percentage of satisfied users.

In this work, we assume that the transmitter sends a fixed set of files during a certain period of time called a carousel cycle and determined by the SF_{opt} as the *SF* value defines the number of packets sent in each cycle. At the end of each cycle, users that cannot download the files and new users that joined the carousel send feedback messages (in the form of received SNR) to the transmitter. Then, the transmitter selects a new MCS mode and *SF* value for the next transmission of the files.

When fountain codes are implemented in the data carousel system, each received symbol will be different and useful for decoding, unlike traditional block codes where the same set of data is transmitted in each cycle. Therefore the calculation of the number of received different symbols in each cycle is required. However, when Raptor codes are implemented, there is no such a constraint so the number of received symbols/packets depends only on the PER. Based on this, next we formulate the download time for the carousel system which is combined with the fountain codes.

First, the expected number of carousel cycles is calculated from the estimated number of received symbols per source block per cycle. The number of received packets per source block depends on the PER encountered in that source block, $PER_l,\ l = 1, \ldots, N_{SB}$. These PER values can be derived either using a Markov model, or from detailed PHY layer simulations. The number of correctly received packets N_r, for each source block *l* at cycle *i* is given by

$$N_{r,l}^{i} = \lfloor (1 - PER_l^i).k.SF_i \rfloor, \qquad l = 1, \ldots, N_{SB} \qquad (10)$$

The number of required carousel cycles can be estimated for each source block as

$$N_{cyc,l} = \min\{C: \textstyle\sum_{i=1}^{C} N_{r,l}^{i} \geq k + \alpha\} \qquad (11)$$

A user may download the files before the end of a carousel cycle since the number of transmitted packets in one cycle may be higher than the number of original source packets or users might receive a number of packets in the previous cycle(s), if the number of cycle is higher than one. Therefore, we calculate the number of received packets in the last cycle, which we call the remaining packets N_{rem}.

$$N_{rem,l} = k + \alpha - \textstyle\sum_{i=1}^{N_{cyc,l}-1} N_{r,l}^{i} \qquad (12)$$

The total number of packets needed for a source block for successful decoding is

$$N_{tot,l} = \textstyle\sum_{i=1}^{N_{cyc,l}-1} k.SF_i + N_{rem,l} \qquad (13)$$

Finally, the source block sb, which requires the highest number of packets for successful decoding determines the download time of a file, i.e. the last decoded source block.

$$sb = \arg\max_{l} N_{tot,l} \qquad (14)$$

The download time is calculated as

$$T_D^{m_j} = \textstyle\sum_{i=1}^{N_{cyc,sb}-1} (N_{SB}.N_F.k.SF_i.Tx^{m_j} + T_{fb}) + (N_{SB}.N_F.(N_{rem,sb} - 1) + 1).Tx^{m_j} + T_{EW} \qquad (15)$$

where T_{fb} is the time to transmit the feedback message, which is assumed 100 bytes[4], with the lowest MCS mode (MCS 0), T_{EW} is the wait time which takes the value between 0 and T_W, so the expected value will be $T_{EW} = T_w/2$. If users want to download all the files, $T_{EW} = 0$.

IV. RESULTS AND ANALYSIS

The performance of multicast file delivery over WLANs is analysed in terms of average download time over all users and percentage of satisfied users. First, we evaluate the *SF* thresholds. Then, we investigate the performance of the adaptive system in a realistic environment. It should be noted that as 100 users were placed at random locations, we performed 20 independent simulations with different topologies and the following results were averaged over all these simulations.

Fig. 7 shows the average download time with respect to the *SF* threshold. Note that for purposes of comparison, the same *SF* threshold is used for each MCS mode during the calculations of the average download time. The MCS mode and SF_{opt} were selected in each cycle using the methodology explained in Section III A. It can be observed that increasing the *SF* threshold results in a longer download time. As explained before increasing the *SF* threshold causes longer delay for the users with bad channel conditions. However, sending less encoded symbols also increases the download time since the users that have not downloaded files have to wait for the next cycle to have a chance to download the file(s).

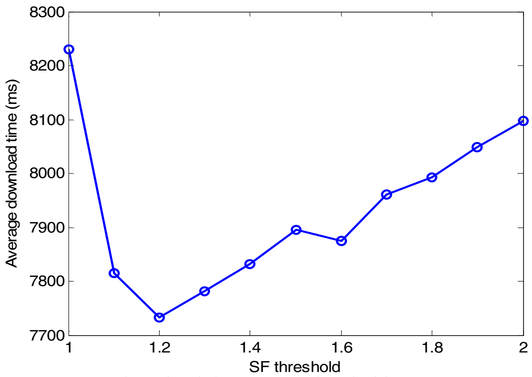

Fig. 7. Average download time vs. *SF* threshold.

Fig. 8 shows the percentage of users that successfully downloaded the files in the first cycle depending on the *SF* threshold. It can be seen that increasing the *SF* threshold significantly increases the percentage of satisfied users in the first cycle and hence reduces the number of users that will send feedback messages to the server, e.g. sending 10% of more encoded packets reduced the number of feedback messages by 61%.

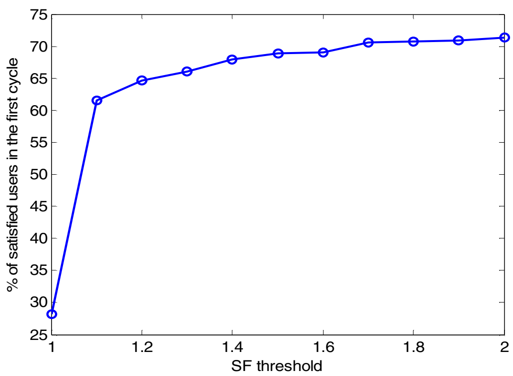

Fig. 8. Percentage of satisfied users in the first cycle vs. *SF* threshold.

Fig. 9 shows the number of required carousel cycles with the *SF* threshold. The results were averaged over all users and realisations. As seen, with the increase of *SF* threshold, the number of required carousel cycles is decreased. However, reducing the number of cycles does not reduce the average download time due to the increase in the period of the carousel.

The results suggest that for an adaptive FEC carousel system, it is important to define the *SF* threshold carefully in order to reduce the average download time and the feedback traffic.

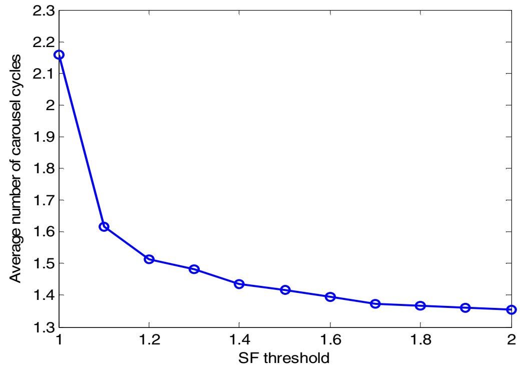

Fig. 9. Number of carousel cycles vs. *SF* threshold.

Next, we evaluated the proposed system performance using the optimum *SF* threshold values in Table II. Fig. 10 compares the percentage of satisfied users versus the download time for the adaptive FEC carousel and the conventional interleaved FEC carousel system where the MCS mode is fixed and *SF* does not affect the system performance (i.e. for the traditional interleaved Raptor AL-FEC carousel, the only parameter that can be configured is MCS mode [7]). The results indicate that lower MCS modes provide greater coverage (percentage of satisfied users) at the expense of longer download times. On the other hand, higher MCS modes provide faster download at the expense of lower coverage. There is a trade-off between coverage and data download time when MCS mode is fixed. However, the adaptive system first selects a higher MCS mode in order to provide faster download for users with good channel conditions and then switches to a lower MCS in order to also provide service to the users with bad channel conditions.

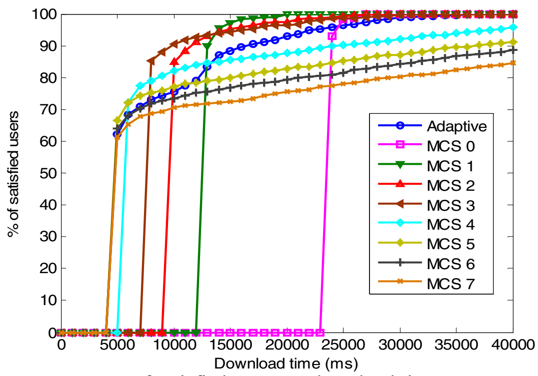

Fig. 10. Percentage of satisfied users vs. download time.

Fig. 11 shows the percentage of serviced users and their average download times. It should be noted that in order to analyse the behaviour of the proposed system, we assume users are continuously moving within the coverage area. As a result, even higher MCS modes provide very high coverage, i.e. all modes provide more than 95% of coverage, since the channel conditions of the users are changing with time (users with bad channel conditions can move to better places). This may not be the case in reality, for example, users can be static or spend long time at some locations. In that case the percentage of covered users can be much lower compared to a mobile scenario as seen in [7]. It can be seen from the figure below that the adaptive system minimises the average download time while it provides coverage for all users. When we compare the adaptive system and the fixed MCS mode transmission, there is a reduction in the average download time between 16 and 67% depending on the MCS chosen.

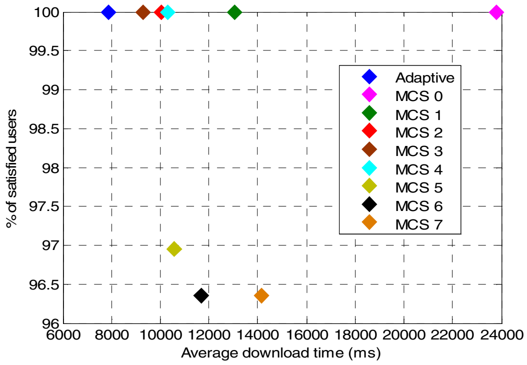

Fig. 11. Percentage of satisfied users vs. average download time.

Finally, we compared the transmission efficiency in terms of total transmission time. Fig. 12 shows the normalised total transmission time which was calculated as the total channel occupancy time from the start of the transmission to the last download (the last user who successfully downloaded the files). That in turn depends on the number of required carousel cycles. As seen, MCS 1 uses less radio and network resources since it requires on average two carousel cycles, contrary to the adaptive system which requires on average three cycles since depending on the distributions of the users PER, it generally switches to a high, moderate and low MCS mode respectively. However, the adaptive system still requires lower channel occupancy time as compared to all other MCS modes (MCS 0, 2-7) since changing the MCS based on the users channel conditions increases the bandwidth efficiency. When MCS index increases the total channel occupancy time increases due to the higher PER values at the higher modes, that results in the number of required carousel cycles to increase. Users with bad channel conditions reduce the overall system performance when the MCS mode is fixed.

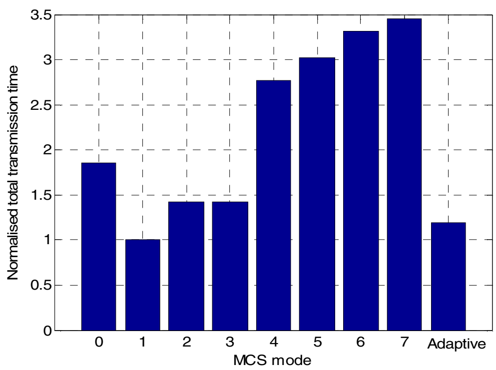

Fig. 12. Normalised total transmission (channel occupancy) time with each MCS mode and adaptive system.

V. Conclusions

In this paper an adaptive Raptor AL-FEC enabled data carousel was proposed to provide reliable and scalable multicast transmissions over 802.11 WLANs. The design problem was formulated and then extensive simulations were performed to evaluate the system performance. The simulation results were presented for a complex real-world environment (Bristol Zoo). The average download time and percentage of satisfied users were used as key design metrics. Results have shown that the adaptive system significantly reduces the average download time, increases the percentage of satisfied users and effectively utilise the valuable radio and network resources.

Acknowledgments

The authors would like to thank the Technology Strategy Board (TSB) for part-funding this work as part of the Arkive In the Your Pocket (AIYP) project. They would also like to thank Bristol Zoo for hosting the Wi-Fi multicast multimedia user trials.

References

[1] S. Acharya, *PhD Thesis*, “Broadcast Disks: Dissemination-based Data Management for Asymmetric Communication Environments”, Brown University, September 1997.

[2] M. Chipeta, K. Karaliopoulos, B.G. Evans, R. Tafazolli, ”On the use of packet-level FEC and data carousels for the delivery of broadcast/multicast services to mobile terminals”. *VTC 2005*, Sweden, May/June 2005; 4:2394–99.

[3] J. Peltotalo, S. Peltotalo, J. Harju, and R. Walsh, “Performance Analysis of a File Delivery System Based on the FLUTE Protocol”. *International Journal of Communication Systems*, October 2006.

[4] I. de Fez and J. C. Guerri, "An Adaptive Mechanism for Optimal Content Download in Wireless Networks," *IEEE Trans. Multimedia* , vol.16, no.4, pp.1140-55, Jun. 2014.

[5] A. Shokrollahi, “Raptor codes”, *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2551–2567, June 2006.

[6] B. Bulut, A. Doufexi, A. Nix, "Carousel-based wireless multicast transmission over WLANs employing raptor codes," *PIMRC,* Sept. 2013.

[7] B. Bulut, D. Berkovskyy, D. Kong, A. Doufexi, A. Nix, “A Raptor Enabled Data Carousel for Enhanced File Delivery and QoS in 802.11 Multicast Networks”, in *IEEE Wireless Communications and Networking Conference, WCNC,* 9-12 March 2015.

[8] M. Luby, A. Shokrollahi, M. Watson, T. Stockhammer, and L. Minder, “RaptorQ Forward Error Correction Scheme for Object Delivery,” RFC 6330, *Internet Engineering Task Force*, Aug. 2011.

[9] B. Bulut, D. Berkovskyy, V. Sgardoni, A. Doufexi and A. Nix, “Robust Multicast Delivery for WLANs: Transforming Theory into Practice”, in *International Conference on Communication, Management and Information Technology (ICCMIT),* 20-22 April 2015.

[10] K. H. Ng, E.K. Tameh, A. Doufexi, M. Hunukumbure, A. R. Nix, “Efficient Multielement Ray Tracing With Site-Specific Comparisons Using Measured MIMO Channel Data,” *IEEE Trans, Veh. Tech*, vol. 56, no. 3, pp. 1019-32, May 2007.

[11] L. Wan, S. Tsai, and M. Almgren, “A fading-insensitive performance metric for a unified link quality model,” in *IEEE Wireless Communications and Networking Conference, WCNC*, vol. 4, April 2006.

[12] IEEE 802.11n Working Group, "IEEE Standard for Information Technology-Telecoms and Information Exchange Between Systems-Local and Metropolitan Area Networks-Specific Requirements-Part 11: WLAN MAC and PHY Specification-Amendment 5: Enhancements for Higher Throughput", Sept. 2009.